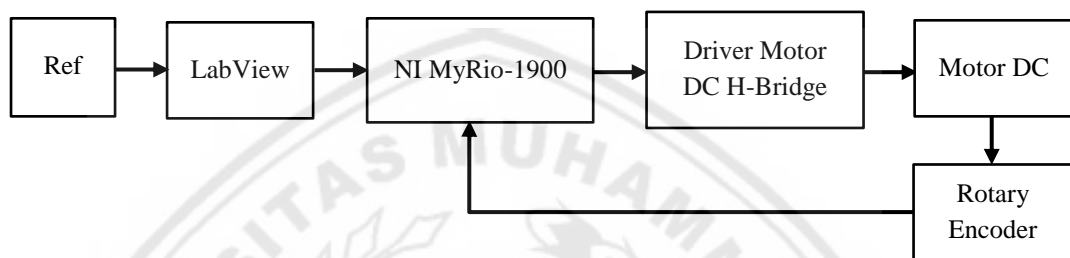


### BAB III

#### PERANCANGAN DAN PEMBUATAN SISTEM

Pada bab ini menjelaskan tentang perancangan dan pembuatan sistem kontrol, baik secara *software* maupun *hardware* yang digunakan untuk mendukung keseluruhan sistem yang akan diuji. *Hardware* beberapa bagian yang saling terhubung dalam sistem, sedangkan *software* digunakan sebagai perantara antara computer dan *hardware*. Pada sistem juga terdiri dari beberapa komponen pendukung yang ditunjukkan pada Gambar 3.1.



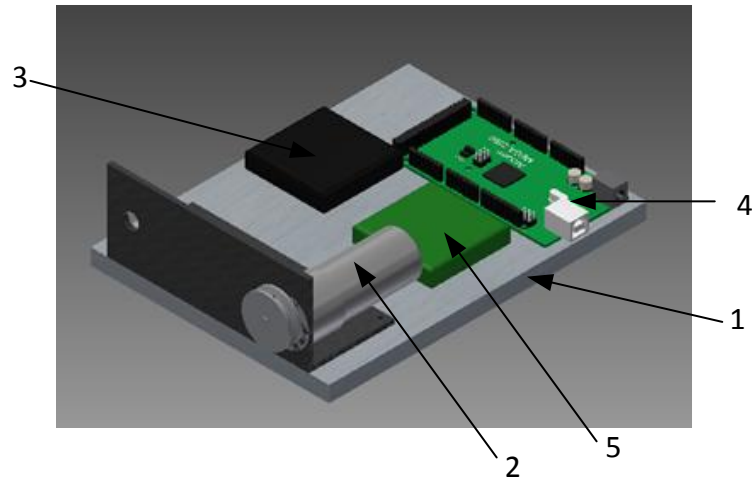
Gambar 3.1 Blok diagram perancangan sistem

Prinsip kerja dari sistem ini adalah sinyal ref memberikan perintah ke *software* LabView yang akan mengirim data ke NI MyRio-1900. NI MyRio-1900 sebagai penghubung ke driver motor DC agar motor DC dapat berputar. *Rotary Encoder* digunakan sebagai sensor kecepatan motor dan mengirim data ke NI MyRio yang nantinya akan menampilkan respon dari *plant*.

#### 3.1 Perancangan Mekanik

Perancangan dan pembuatan mekanik akan dibuat dengan spesifikasi seperti pada Gambar 3.2.

1. Menggunakan papan kayu dengan:
  - Ketebalan 15 mm
  - Panjang 370 mm
  - Lebar 260 mm
2. Motor DC + *rotary encoder*
3. Power suplay
4. NI MyRio-1900
5. Driver motor

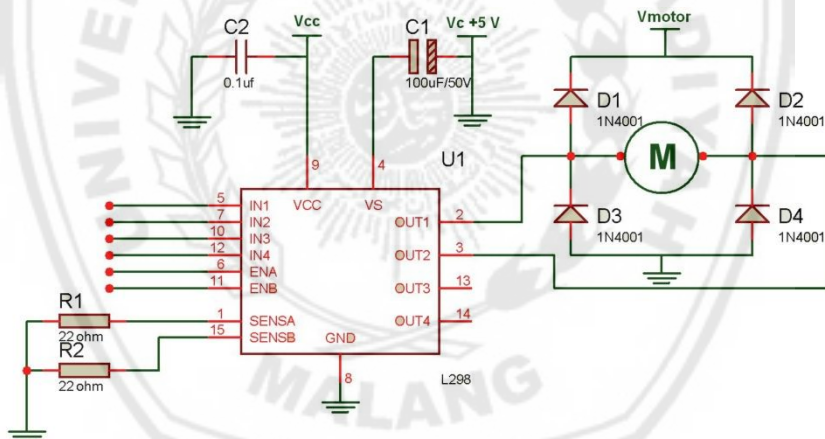


Gambar 3.2 Perancangan Mekanik

### 3.2 Perancangan *Hardware*

#### 3.2.1 Perancangan *Driver Motor DC*

Driver motor dalam skripsi ini merupakan implementasi dari prinsip H-Bridge (jembatan H) untuk mengontrol kecepatan serta arah perputaran motor DC.

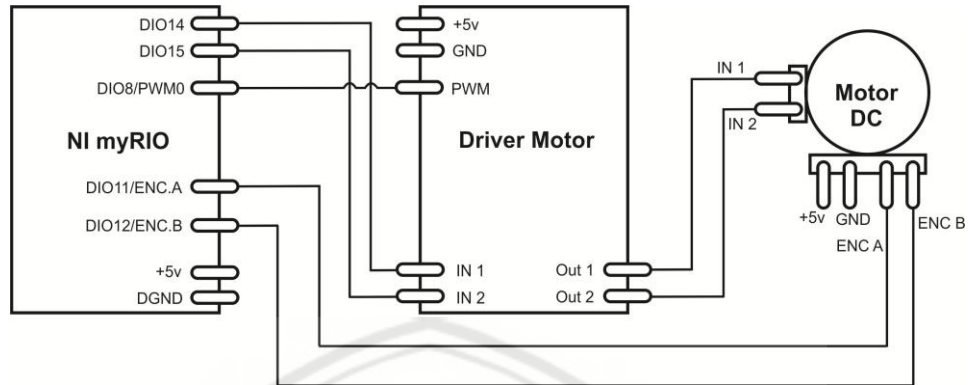


Gambar 3.3 Rangkaian *Driver Motor DC*

- PIN 9 Vcc 12 Volt
- PIN 4 Vc +5 Volt
- PIN 1 dan PIN 2 dihubungkan ke Motor DC
- PIN 5 dan PIN 7 dihubungkan ke PIN Digital Input Output pada NI myRIO
- PIN 6 dihubungkan ke PIN PWM pada NI myRIO untuk mengontrol kecepatan motor DC

### 3.2.2 Perancangan *Hardware* Sistem

Pada pembuatan modul sistem control, diperlukan rangkaian pendukung hardware yang ditunjukkan pada Gambar 3.3.



Gambar 3.4 Perancangan Hardware

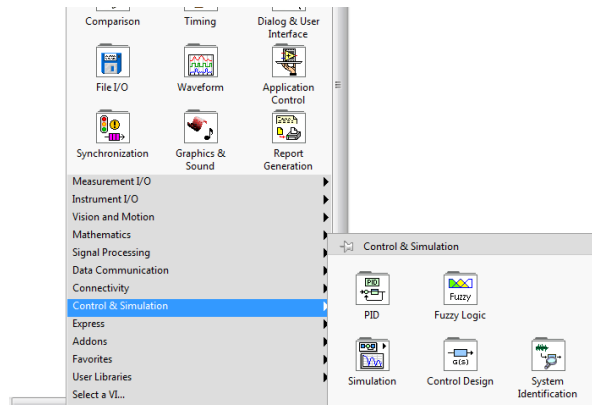
Keterangan:

- PIN DIO14 NI myRIO dihubungkan ke PIN IN 1 Driver motor
- PIN DIO15 NI myRIO dihubungkan ke PIN IN 2 Driver motor
- PIN DIO8/PWM NI myRIO dihubungkan ke PIN PWM Driver motor
- PIN DIO11/ENC A NI myRIO dihubungkan ke PIN Rotary Encoder A
- PIN DIO12/ENC B NI myRIO dihubungkan ke PIN Rotary Encoder B
- PIN Out 1 Driver motor dihubungkan ke IN 1 Motor DC
- PIN Out 2 Driver motor dihubungkan ke IN 2 Motor DC

### 3.3 Perancangan Sistem Pada LabVIEW

#### 3.3.1 Identifikasi Motor DC

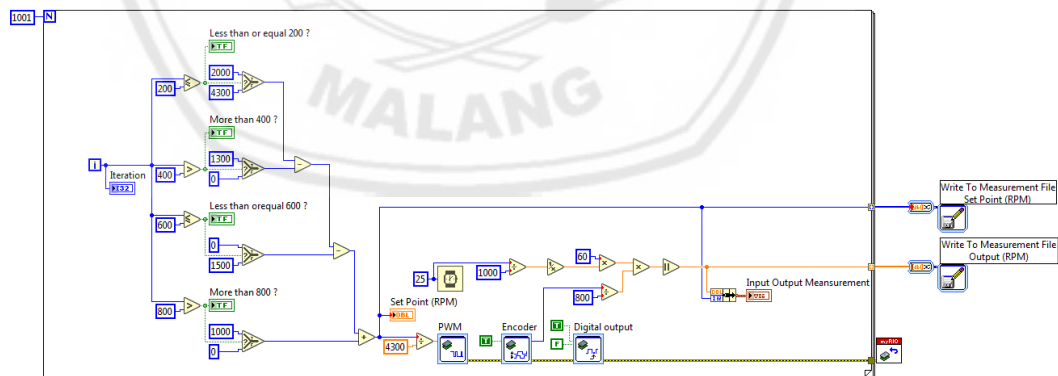
Pemodelan sistem motor DC menjadi bagian utama pada sistem ini karena akan dikontrol menggunakan pengaturan PID. Pemodelan ini dilakukan menggunakan *system identification toolbox* yang merupakan fitur dari *software LabVIEW*. Beberapa langkah yang dilakukan dalam pemodelan sistem adalah :



Gambar 3.5 Identification System Toolkit pada LabVIEW

### 3.3.1.1 Pengambilan Data

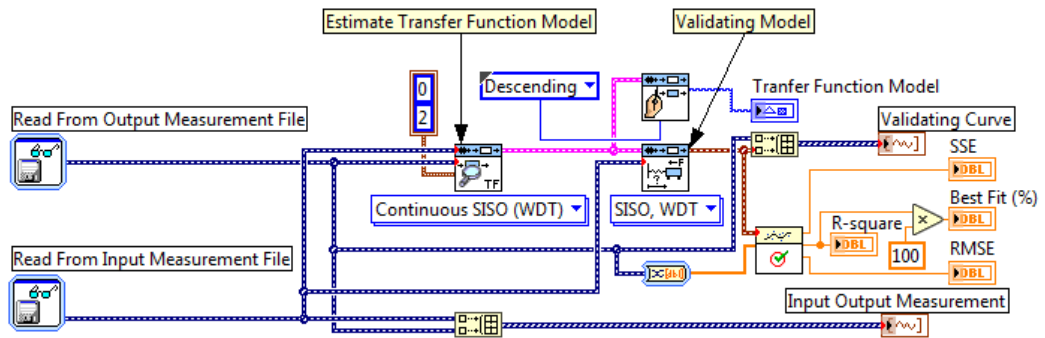
Data input yang diberikan pada motor DC adalah *setpoint* kecepatan motor yang telah dirancang. Perancangan *setpoint* yang dipakai untuk data input dirancang menggunakan salah satu *tools* pada LabVIEW. Data *output* yang dipakai adalah data kecepatan motor yang didapat dari pembacaan sensor kecepatan pada motor DC. Pengujian dilakukan selama 25 detik dan setiap 5 detik amplitudo *signal setpoint* akan berubah dengan *sampling time* 25ms. Tujuan pemilihan *time sampling* 25ms agar sesuai aplikasi lain yang membutuhkan kontrol cepat seperti tekanan atau laju aliran (*High-Speed Digital Controller*). Gambar di bawah menunjukkan rancangan *Listing Program* pengujian *input-output* untuk proses estimasi parameter.



Gambar 3.6 Graphical Program pengujian Input-Output pada LabVIEW

### 3.3.1.2 Estimasi dan Validasi

Validasi dilakukan untuk mengetahui tingkat keakuratan pemodelan motor DC terhadap *plant* Motor DC sesungguhnya (*hardware*) dalam bentuk presentase sering juga disebut *best-fit*.



Gambar 3.7 Graphical Program Estimasi Parameter dan Validasi  
Pemodelan Motor DC pada LabVIEW

Nilai *best-fit* dihitung menggunakan *R-square*, semakin mendekati 1 maka semakin baik tingkat keakuratannya. Sedangkan RMSE adalah akar-akar kesalahan kuadrat, dapat diperoleh dengan persamaan di bawah ini.

$$SSE = \sum_{i=0}^{n-1} w_i(y_i - f_i)^2 \quad (3.1)$$

$$SST = \sum_{i=0}^{n-1} w_i(y_i - \bar{y})^2 \quad (3.2)$$

$$RMSE = \sqrt{\frac{SSE}{DOF}} \quad (3.3)$$

$$R\ Square = 1 - \frac{SSE}{SST} \quad (3.4)$$

$$Best\ fit = R\ Square \times 100\% \quad (3.5)$$

Dengan:

*SSE* : jumlah kesalahan kuadrat (*summation of square error*)

*SST* : total jumlah kuadrat (*total sum of square*)

*RMSE* : akar rata-rat kesalahan kuadrat (*root mean squared error*)

*DOF* : derajat kebebasan (*degree of freedom*)

*y<sub>i</sub>* : elemen ke-*i* dari *y*

$\bar{y}$  : rata-rata dari *y*

*f<sub>i</sub>* : elemen ke-*i* dari *best-fit*

*w<sub>i</sub>* : elemen ke-*i* dari bobot

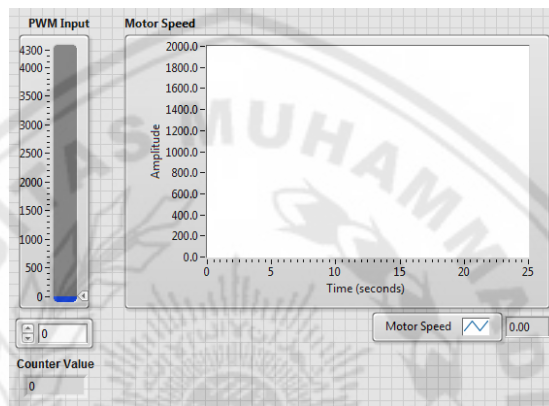
*SST* : *SSR* + *SSE*

*R Square*: perbandingan antara SSR (*sum of squares of theregression*) dan SST (*total sum of square*)

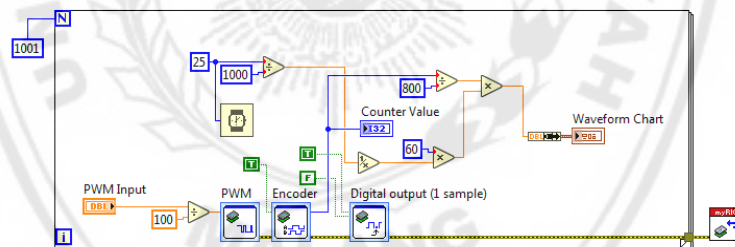
*Best fit*: tingkat keakuratan pemodelan (%)

### 3.3.2 Perancangan Sistem *Openloop*

Gambar 3.7 di bawah ini menunjukkan tampilan untuk sistem *openloop* yang akan digunakan untuk menentukan parameter untuk kontrol PID dengan cara memberikan *input* yang berupa *setpoint* tertentu. Dari hasil pengujian akan didapatkan kurva, kemudian dari kurva tersebut akan didapatkan parameter L dan parameter T.



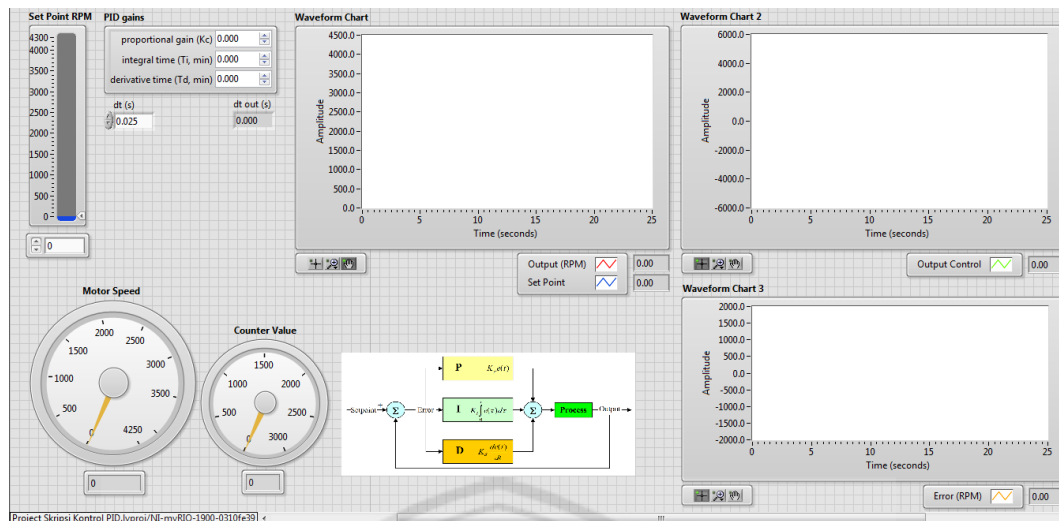
Gambar 3.8 Perancangan *Front Panel* Sistem *Openloop* pada LabVIEW



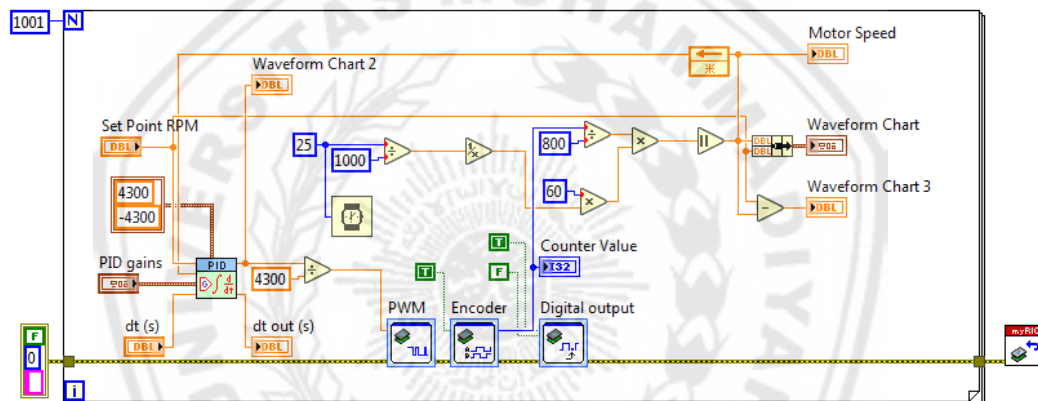
Gambar 3.9 *Graphical Program* Sistem *Openloop* pada LabVIEW

### 3.3.3 Perancangan Sistem *Closeloop* Kontrol PID

Pada Gambar 3.9 menunjukkan tampilan rancangan pada LabVIEW untuk pengontrolan PID terhadap *plant* motor DC yang terdiri dari *input* untuk *set point* RPM, *input* untuk *pid gains*, *gauge Motor Speed*, *Counter Gauge*, serta *waveform graph* untuk menampilkan hasil dari pengontrolan kecepatan putaran motor DC.



Gambar 3.10 Perancangan *Front Panel* Kontrol PID pada LabVIEW



Gambar 3.11 *Graphical Program* Kontrol PID pada LabVIEW

Pada Gambar 3.10 yaitu *graphical program* dari kontrol PID yang dirancang untuk pengontrolan kecepatan putar motor DC, pada gambar di atas dapat dilihat nilai *set point* dimasukan ke dalam *toolkit* PID yang mana di dalamnya telah diproses sebelum dikeluarkan pada plant motor DC.

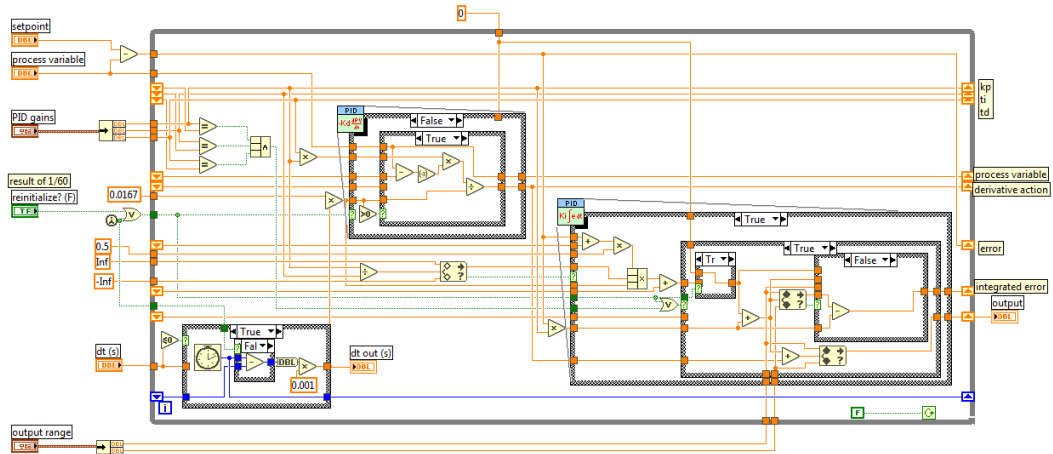
PID kontroler membandingkan *set point* (SP) ke *Present Value* (PV) atau nilai aktual untuk mendapatkan kesalahan (e).

$$\text{Error}(t) = \text{SP} - \text{PV}(t)$$

SP = *Set Point* (Nilai yang diinginkan)

PV(t) = *Present Value* pada waktu(t)

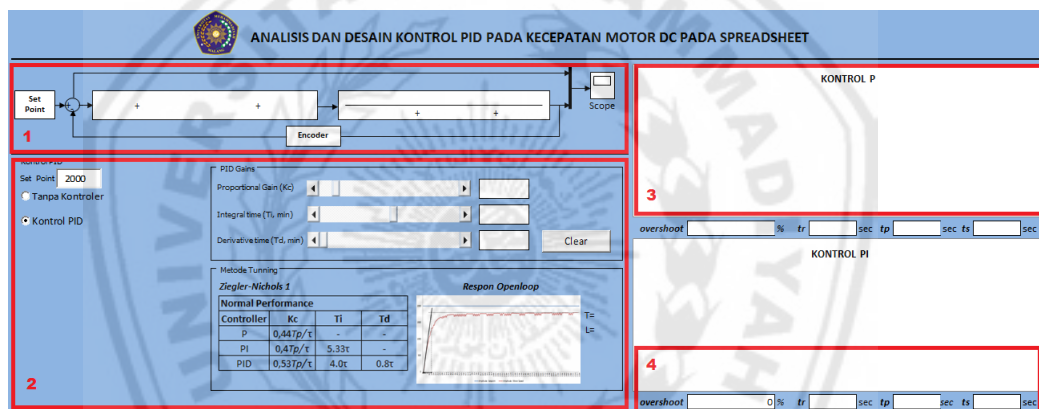




Gambar 3.12 *Graphical Program* dari Toolkit Kontrol PID pada LabVIEW

### 3.4 Perancangan Sistem *Closeloop* Kontrol PID pada *Spreadsheet*

Perancangan sistem pada *spreadsheet* dengan menggunakan bahasa pemrograman VBA yang terdapat pada Microsoft Excel.



Gambar 3.13 Perancangan Sistem pada *Spreadsheet*

Pada Gambar 3.12 menunjukan beberapa bagian dari sistem yaitu pada bagian nomor 1 adalah bagian untuk menampilkan *input* parameter PID serta *plant* motor DC yang didapatkan dari estimasi parameter secara *openloop* dan gambar diagram blok sistem kontrol. Bagian nomor 2 menunjukkan *User Interface* (UI) yang digunakan untuk memasukkan *input setpoint* putaran motor DC dalam satuan rotasi per menit (RPM), *OptionBox* untuk pemilihan metode pengontrolan, “Tanpa Kontroler” dan “Kontrol PID”. Sedangkan pada bagian nomor 3 yaitu gelombang yang dihasilkan dari proses analisa yang dilakukan dan nomor 4 adalah respon sistem. Pada perancangan sistem pada *spreadsheet* terdapat beberapa *sheet* diantaranya adalah *sheet* “Interface” yang digunakan sebagai *User Interface* (UI) sebagaimana yang telah dijelaskan di atas, sedangkan pada *sheet* “Sistem Kontrol” digunakan untuk mengolah blok diagram kontrol dan *plant*



motor DC, dan *sheet* “Runge-Kutta 2 (Midpoint)” digunakan untuk penyelesaian persamaan diferensial.

Perancangan kontroler PID terhadap *plant* sistem motor DC dengan cara memasukkan parameter kontrol PID pada diagram blok dalam bentuk masing-masing fungsi alih. Kemudian mengolah blok diagram kontrol PID dan sistem motor DC secara seri (konvolusi), maupun umpan balik (*feedback*). Setelah mendapatkan hasil konvolusi dalam bentuk fungsi alih, maka selanjutnya akan diubah ke persamaan diferensial dan menyelesaikan persamaan tersebut menggunakan *Runge-Kutta* orde 2.

Karena kita sudah memiliki persamaan (2.7) dengan empat nilai yang belum diketahui yaitu  $a_1$ ,  $p_1$  dan  $q_{11}$ , untuk mendapatkan nilai tersebut kita harus mengasumsikan nilai dari salah satu yang tidak diketahui menggunakan persamaan (2.8) dan persamaan (2.9).

Pada penelitian ini, untuk menentukan nilai dari  $a_1$ ,  $p_1$  dan  $q_{11}$  digunakan metode titik tengah (*Midpoint*) yaitu dengan mengasumsikan nilai dari  $a_2 = 1$  maka akan didapatkan  $a_1 = 0$  dan  $p_1 = q_{11} = \frac{1}{2}$ , persamaan (2.7) menjadi:

$$\begin{aligned} k_1 &= hf(x_i, y_i) \\ k_2 &= hf\left(x_1 + \frac{1}{2}h, y_i + \frac{1}{2}k_1h\right) \\ y_{i+1} &= y_i + k_2h \end{aligned} \tag{3.6}$$